

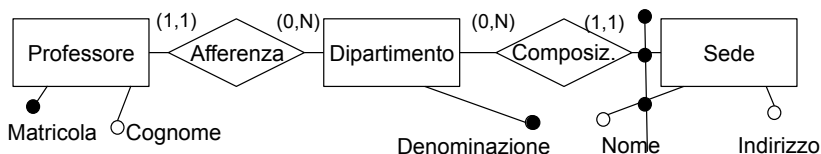
Possibili soluzioni

Tempo a disposizione: un'ora e quarantacinque minuti.

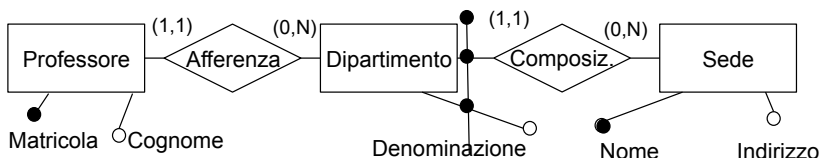
Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Domanda 1 (10%) Considerare i due schemi seguenti

a)



b)



Con riferimento a ciascuno di essi, rispondere alle seguenti domande, con un sì o un no negli spazi della tabella:

riordinare nei compiti B, C, D

	Schema (a)	Schema (b)
possono esistere due sedi con lo stesso nome per uno stesso dipartimento	NO	NO
possono esistere due sedi per un dipartimento?	SÌ	NO
possono esistere due dipartimenti per una sede?	NO	SÌ
possono esistere due sedi con lo stesso nome?	SÌ	NO
ad un professore è associato uno e un solo indirizzo?	NO	SÌ

Domanda 2 (15%) Tradurre nel modello relazionale i due schemi ER mostrati nella domanda precedente, mostrando possibili istanze (che rendano evidenti le differenze fra le due realtà; bastano tre o quattro ennuple per ciascuna relazione). Indicare (ad esempio direttamente sulle tabelle) anche i vincoli di chiave e integrità referenziale.

a)  
 (scambiare con la successiva nei compiti B e D)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip )  
 - DIPARTIMENTO (Denominazione, ... )  
 - SEDE (DenominazioneDip, NomeSede, Indirizzo ... )

b)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip, SedeDip )  
 - DIPARTIMENTO (Denominazione, Sede ... )  
 - SEDE (NomeSede, Indirizzo ... )

**Domanda 3** (15%) Considerare il seguente schema relazionale (con vincoli di integrità referenziale fra l'attributo Prodotto di LINEEDORDINE e la chiave Matricola di PRODOTTI e fra l'attributo Ordine di LINEEDORDINE e la chiave Codice di ORDINI):

PRODOTTI			LINEEDORDINE			ORDINI		
Matricola	Nome	Prezzo	Prodotto	Ordine	Quantità	Codice	Data	Cliente
...	...	...	...	...	...	...	...	...

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità  $P = 10.000$  (PRODOTTI),  $O = 1.000$  (ORDINI) e  $L = 3.000$  (LINEEDORDINE), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT Matricola, Codice FROM Prodotti, Ordini	$P \times O$	$P \times O$	10.000.000	10.000.000
SELECT Prodotto, Ordine, Nome, Data FROM Prodotti JOIN LineeDOrdine ON Matricola=Prodotto JOIN Ordini ON Ordine = Codice	$L$	$L$	3.000	3.000
SELECT Prodotto, Nome, COUNT(*) FROM Prodotti JOIN LineeDOrdine ON Matricola=Prodotto GROUP BY Prodotto, Nome	$L/O$	$P$	3	10.000

Nota: Il valore  $L/O$  deriva dal numero minimo di prodotti che deve essere ordinato. Il valore deriva dal fatto che ogni ordine ha linee che riguardano prodotti diversi.

**Domanda 4** (25%) Considerare ancora lo schema mostrato nella domanda precedente.

- Formulare in **SQL** l'interrogazione che trova le coppie di prodotti (è sufficiente mostrare gli identificatori) che compaiono insieme in almeno un ordine.

```
SELECT DISTINCT L1.Prodotto AS Prodotto1, L2.Prodotto AS Prodotto2
FROM LineeDOrdine L1, LineeDOrdine L2
WHERE L1.Ordine = L2.Ordine AND L1.Prodotto > L2.Prodotto
```

$$\pi_{P,P'}(\sigma_{P>P'}(\text{LINEEDORDINE} \bowtie_{O=O'} \rho_{X' \leftarrow X}(\text{LINEEDORDINE})))$$

- Formulare in **algebra relazionale** l'interrogazione che trova i clienti che hanno ordinato almeno due prodotti (in uno stesso ordine o in ordini diversi). Mostrare, oltre ai clienti, le coppie di prodotti

Utile una vista

```
CREATE VIEW V AS SELECT *
FROM LineeDOrdine JOIN Ordini ON Ordine = Codice
```

```
SELECT V1.Cliente, V1.Prodotto, V2.Prodotto
FROM V AS V1 JOIN V AS V2 ON V1.Cliente = V2.Cliente
WHERE V1.Prodotto > V2.Prodotto
```

$$V := \text{LINEEDORDINE} \bowtie_{O=C} \text{ORDINI}$$

$$\pi_{P,P'}(\sigma_{P>P'}(V \bowtie_{C=C'} \rho_{X' \leftarrow X} V))$$

- Formulare in **SQL** l'interrogazione che trova i prodotti (mostrando tutte le proprietà) che sono stati ordinati da un solo cliente (in uno o più ordini).

```
SELECT Prodotti.*
FROM Prodotti JOIN V V1 ON Matricola = Prodotto
WHERE NOT EXISTS (SELECT *
FROM V V2
WHERE V1.Cliente <> V2.Cliente AND V1.Prodotto = V2.Prodotto)
```

La domanda continua a pagina seguente.

Ripetiamo per comodità lo schema:

PRODOTTI			LINEEDORDINE			ORDINI		
Matricola	Nome	Prezzo	Prodotto	Ordine	Quantità	Codice	Data	Cliente
...	...	...	...	...	...	...	...	...

4. Formulare in SQL l'interrogazione che trova tutti gli ordini, indicando, per ciascuno, data, cliente, numero di prodotti diversi ordinati e importo totale (ottenuto attraverso la somma dei prodotti fra quantità e prezzo)

```
SELECT Codice, Data, Cliente, COUNT(*) AS NProdotti, SUM(Quantita*Prezzo)
FROM V JOIN Prodotti ON Prodotto = Matricola
GROUP BY Codice, Data, Cliente
```

**Domanda 5** (7.5%) Con riferimento ancora allo schema usato nelle domande precedenti, si vuole scrivere un metodo che stampi l'elenco dei prodotti, specificando per ciascuno l'elenco degli ordini in cui compare, con il cliente che lo ha ordinato e la relativa quantità. Si vogliono stampare anche i prodotti per i quali non c'è nessun ordine. Il frammento di codice che (una volta completato con l'interrogazione SQL passata come argomento alla `executeQuery` nella seconda riga) produce la stampa è il seguente.

```
Statement query = con.createStatement();
ResultSet result = query.executeQuery("                ");
// La riga precedente va completata con l'interrogazione SQL
int prodottoPrec = -1;
while (result.next()){
    int prodotto = result.getInt("prodotto");
    if (!(prodotto==prodottoPrec)) {
        System.out.println("Prodotto: " + prodotto);
        prodottoPrec = prodotto ;
    }
    int ordine = result.getInt("ordine");
    int quantita = result.getInt("quantita");
    String cliente = result.getString("cliente");
    System.out.println(" Ordine: " + ordine + " Cliente: " + cliente + ". Quantita:" +
        quantita);
}
```

Scrivere l'interrogazione SQL da inserire come argomento della `executeQuery` nella seconda riga.

```
SELECT Matricola AS Prodotto, Ordine, Cliente, Quantita
FROM Prodotti LEFT JOIN (LineeDOrdine JOIN Ordini ON Ordine = Codice)
    ON Matricola = Prodotto
ORDER BY Matricola, Ordine
```

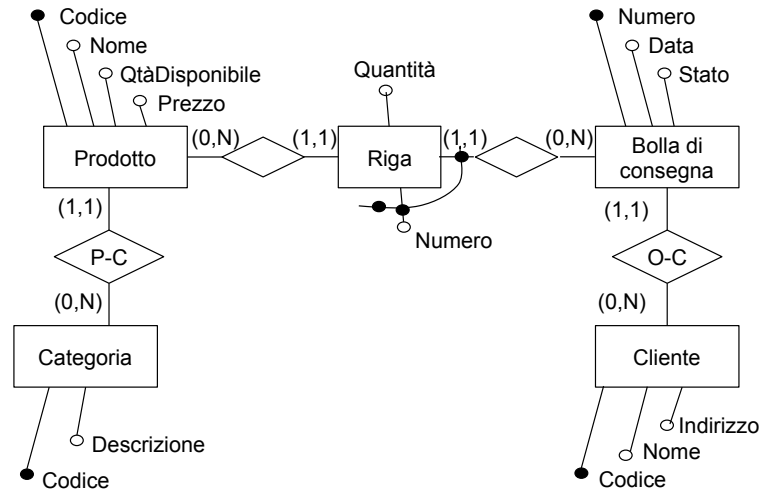
**Domanda 6** (7.5%) Definire (con una opportuna notazione) su una relazione

CONTICORRENTI(Matricola,Saldo,Interessi)

un vincolo che imponga che il valore di **Interessi** è positivo solo se quello di **Saldo** è almeno pari a 5.000.

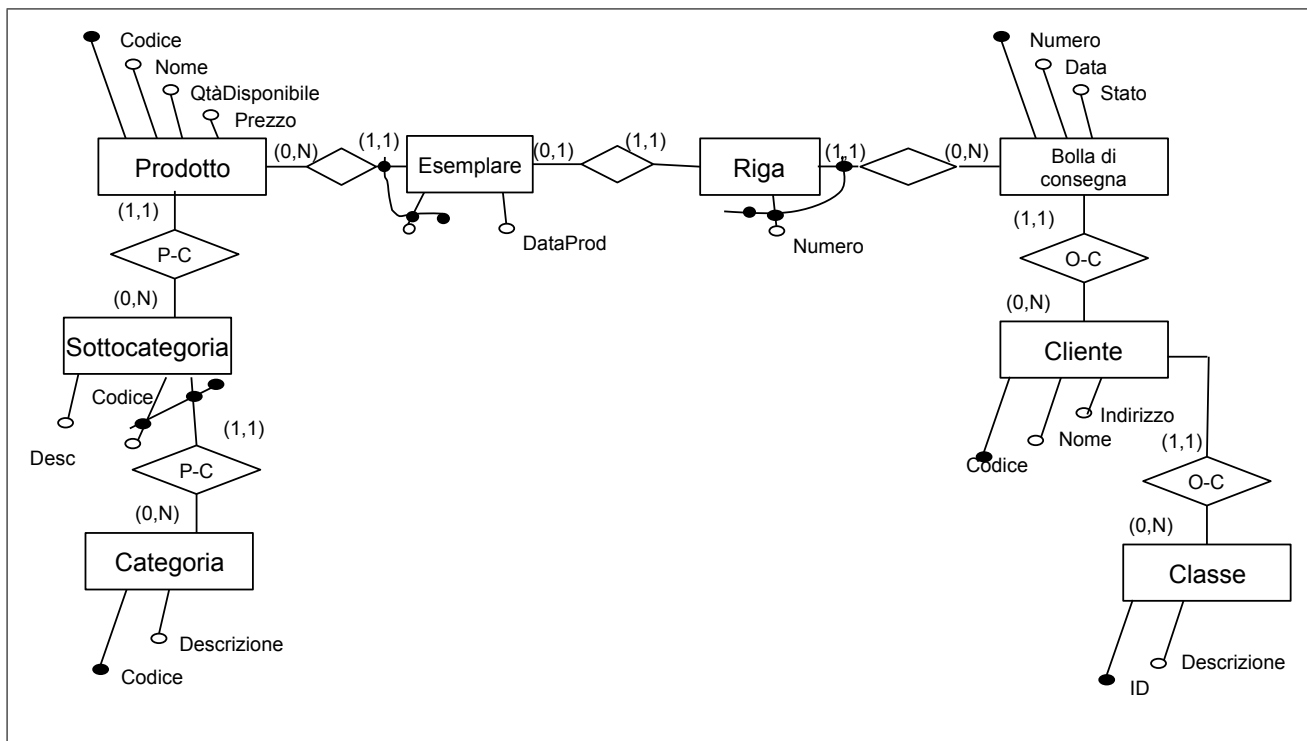
```
CHECK (NOT (Interessi > 0) OR Saldo > 5.000)
```

**Domanda 7 (20%)** Considerare il seguente schema concettuale, simile a quello discusso come soluzione del progetto proposto nel corso (la differenza principale è che, invece che di ordini, si tratta di bolle di consegna, i documenti che descrivono la fase di consegna dei prodotti):



Modificare lo schema sulla base delle specifiche seguenti

- i clienti sono raggruppati in classi (al momento, “privati,” “business” e “top business,” ma potrebbero cambiare nel tempo); ogni classe ha un identificatore e una descrizione;
- le categorie sono articolate in sottocategorie (anche loro con codice e descrizione) e ogni prodotto appartiene ad una sottocategoria; le sottocategorie hanno codici che sono univoci solo nell’ambito di ciascuna categoria;
- per i prodotti sono di interesse (e debbono essere in qualche modo associabili alle rispettive bolle) i singoli esemplari, ognuno dei quali ha un numero di matricola (che identifica l’esemplare fra quelli del prodotto) e una data di produzione



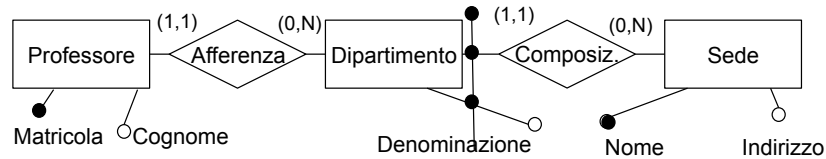
Possibili soluzioni

Tempo a disposizione: un'ora e quarantacinque minuti.

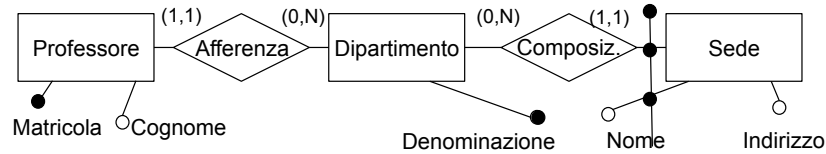
Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Domanda 1 (10%) Considerare i due schemi seguenti

a)



b)



Con riferimento a ciascuno di essi, rispondere alle seguenti domande, con un sì o un no negli spazi della tabella:

riordinare nei compiti B, C, D

	Schema (a)	Schema (b)
possono esistere due sedi per un dipartimento?	NO	NO
possono esistere due sedi con lo stesso nome per uno stesso dipartimento	SÌ	NO
possono esistere due sedi con lo stesso nome?	NO	SÌ
possono esistere due dipartimenti per una sede?	SÌ	NO
ad un professore è associato uno e un solo indirizzo?	NO	SÌ

Domanda 2 (15%) Tradurre nel modello relazionale i due schemi ER mostrati nella domanda precedente, mostrando possibili istanze (che rendano evidenti le differenze fra le due realtà; bastano tre o quattro ennuple per ciascuna relazione). Indicare (ad esempio direttamente sulle tabelle) anche i vincoli di chiave e integrità referenziale.

a)  
 (scambiare con la successiva nei compiti B e D)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip )  
 - DIPARTIMENTO (Denominazione, ... )  
 - SEDE (DenominazioneDip, NomeSede, Indirizzo ... )

b)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip, SedeDip )  
 - DIPARTIMENTO (Denominazione, Sede ... )  
 - SEDE (NomeSede, Indirizzo ... )

**Domanda 3** (15%) Considerare il seguente schema relazionale (con vincoli di integrità referenziale fra l'attributo Prodotto di LINEEDORDINE e la chiave Codice di PRODOTTI e fra l'attributo Ordine di LINEEDORDINE e la chiave Numero di ORDINI):

PRODOTTI			LINEEDORDINE			ORDINI		
Codice	Nome	Prezzo	Prodotto	Ordine	Quantità	Numero	Data	Cliente
...	...	...	...	...	...	...	...	...

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità  $P = 30.000$  (PRODOTTI),  $O = 1.000$  (ORDINI) e  $L = 5.000$  (LINEEDORDINE), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT Codice, Numero FROM Prodotti, Ordini	$P \times O$	$P \times O$	30.000.000	30.000.000
SELECT Prodotto, Ordine, Nome, Data FROM Prodotti JOIN LineeDOrdine ON Codice=Prodotto JOIN Ordini ON Ordine = Numero	$L$	$L$	5.000	5.000
SELECT Prodotto, Nome, COUNT(*) FROM Prodotti JOIN LineeDOrdine ON Codice=Prodotto GROUP BY Prodotto, Nome	$L/O$	$P$	5	30.000

Nota: Il valore  $L/O$  deriva dal numero minimo di prodotti che deve essere ordinato. Il valore deriva dal fatto che ogni ordine ha linee che riguardano prodotti diversi.

**Domanda 4** (25%) Considerare ancora lo schema mostrato nella domanda precedente.

1. Formulare in **algebra relazionale** l'interrogazione che trova le coppie di prodotti (è sufficiente mostrare gli identificatori) che compaiono insieme in almeno un ordine.

```
SELECT DISTINCT L1.Prodotto AS Prodotto1, L2.Prodotto AS Prodotto2
FROM LineeDOrdine L1, LineeDOrdine L2
WHERE L1.Ordine = L2.Ordine AND L1.Prodotto > L2.Prodotto
```

$$\pi_{P,P'}(\sigma_{P>P'}(\text{LINEEDORDINE} \bowtie_{O=O'} \rho_{X' \leftarrow X}(\text{LINEEDORDINE})))$$

2. Formulare in **SQL** l'interrogazione che trova i clienti che hanno ordinato almeno due prodotti (in uno stesso ordine o in ordini diversi). Mostrare, oltre ai clienti, le coppie di prodotti

Utile una vista

```
CREATE VIEW V AS SELECT *
FROM LineeDOrdine JOIN Ordini ON Ordine = Codice
```

```
SELECT V1.Cliente, V1.Prodotto, V2.Prodotto
FROM V AS V1 JOIN V AS V2 ON V1.Cliente = V2.Cliente
WHERE V1.Prodotto > V2.Prodotto
```

$$V := \text{LINEEDORDINE} \bowtie_{O=C} \text{ORDINI}$$

$$\pi_{P,P'}(\sigma_{P>P'}(V \bowtie_{C=C'} \rho_{X' \leftarrow X} V))$$

3. Formulare in **SQL** l'interrogazione che trova i prodotti (mostrando tutte le proprietà) che sono stati ordinati da un solo cliente (in uno o più ordini).

```
SELECT Prodotti.*
FROM Prodotti JOIN V V1 ON Matricola = Prodotto
WHERE NOT EXISTS (SELECT *
FROM V V2
WHERE V1.Cliente <> V2.Cliente AND V1.Prodotto = V2.Prodotto)
```

La domanda continua a pagina seguente.

Ripetiamo per comodità lo schema:

PRODOTTI			LINEEORDINE			ORDINI		
Codice	Nome	Prezzo	Prodotto	Ordine	Quantità	Numero	Data	Cliente
...	...	...	...	...	...	...	...	...

4. Formulare in SQL l'interrogazione che trova tutti gli ordini, indicando, per ciascuno, data, cliente, numero di prodotti diversi ordinati e importo totale (ottenuto attraverso la somma dei prodotti fra quantità e prezzo)

```
SELECT Codice, Data, Cliente, COUNT(*) AS NProdotti, SUM(Quantita*Prezzo)
FROM V JOIN Prodotti ON Prodotto = Matricola
GROUP BY Codice, Data, Cliente
```

**Domanda 5** (7.5%) Con riferimento ancora allo schema usato nelle domande precedenti, si vuole scrivere un metodo che stampi l'elenco dei prodotti, specificando per ciascuno l'elenco degli ordini in cui compare, con il cliente che lo ha ordinato e la relativa quantità. Si vogliono stampare anche i prodotti per i quali non c'è nessun ordine. Il frammento di codice che (una volta completato con l'interrogazione SQL passata come argomento alla `executeQuery` nella seconda riga) produce la stampa è il seguente.

```
Statement query = con.createStatement();
ResultSet result = query.executeQuery(" ");
// La riga precedente va completata con l'interrogazione SQL
int prodottoPrec = -1;
while (result.next()){
    int prodotto = result.getInt("prodotto");
    if (!(prodotto==prodottoPrec)) {
        System.out.println("Prodotto: " + prodotto);
        prodottoPrec = prodotto ;
    }
    int ordine = result.getInt("ordine");
    int quantita = result.getInt("quantita");
    String cliente = result.getString("cliente");
    System.out.println(" Ordine: " + ordine + " Cliente: " + cliente + ". Quantita:" +
        quantita);
}
```

Scrivere l'interrogazione SQL da inserire come argomento della `executeQuery` nella seconda riga.

```
SELECT Matricola AS Prodotto, Ordine, Cliente, Quantita
FROM Prodotti LEFT JOIN (LineeDOrdine JOIN Ordini ON Ordine = Codice)
ON Matricola = Prodotto
ORDER BY Matricola, Ordine
```

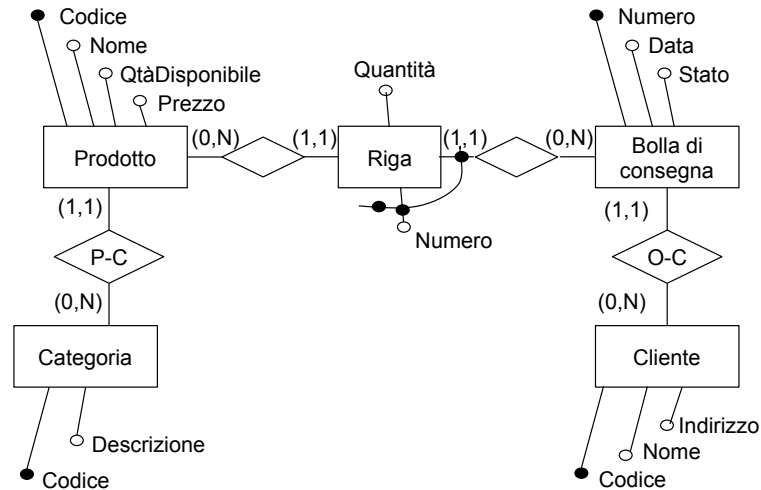
**Domanda 6** (7.5%) Definire (con una opportuna notazione) su una relazione

`SOCI(Matricola,Punti,PuntiPremio)`

un vincolo che imponga che il valore di `PuntiPremio` è positivo solo se quello di `Punti` è almeno pari a 1.000.

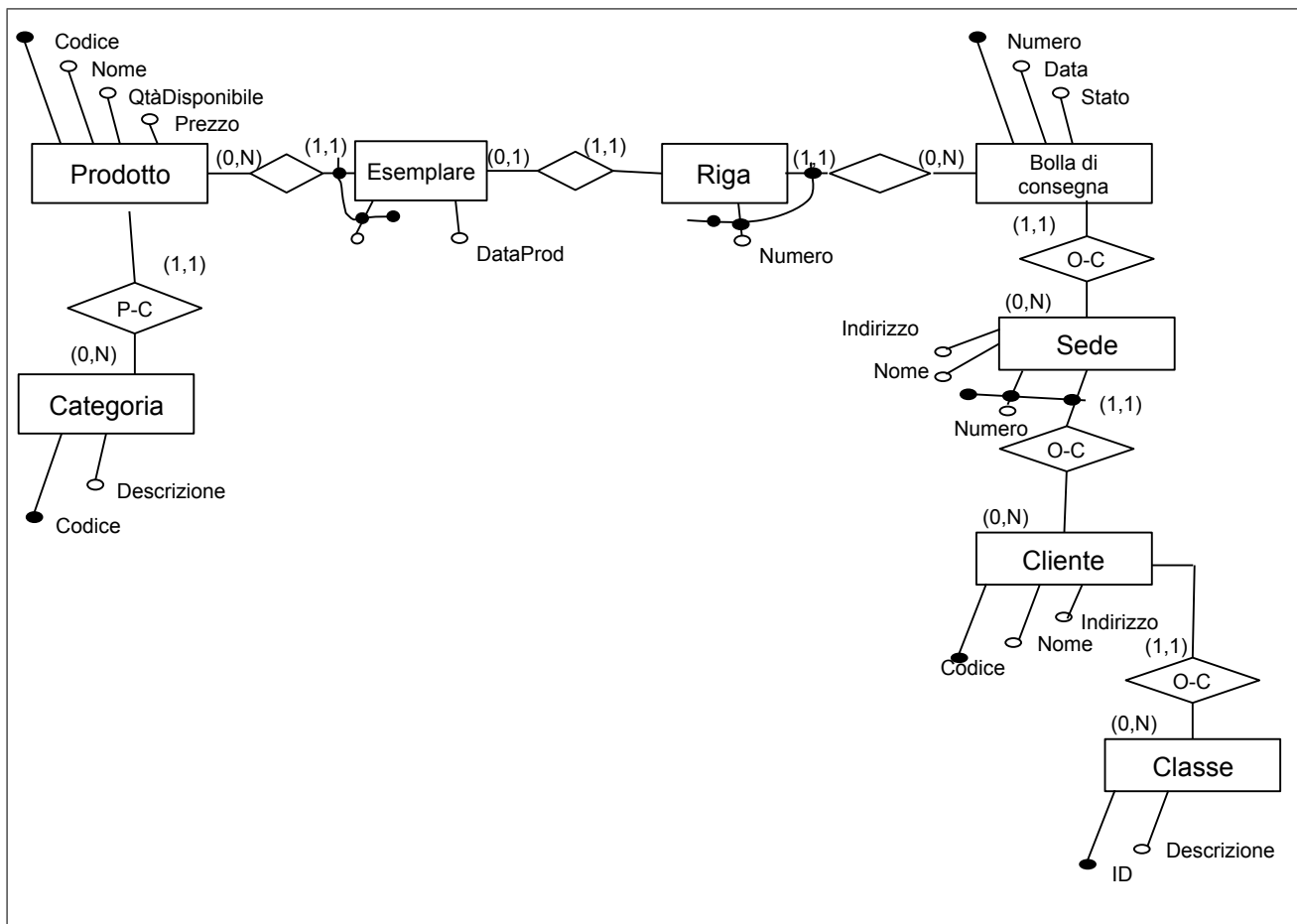
```
CHECK (NOT (PuntiPremio > 0) OR Punti > 1.000)
```

**Domanda 7 (20%)** Considerare il seguente schema concettuale, simile a quello discusso come soluzione del progetto proposto nel corso (la differenza principale è che, invece che di ordini, si tratta di bolle di consegna, i documenti che descrivono la fase di consegna dei prodotti):



Modificare lo schema sulla base delle specifiche seguenti

- i clienti sono raggruppati in classi (al momento, “privati,” “business” e “top business,” ma potrebbero cambiare nel tempo); ogni classe ha un identificatore e una descrizione;
- ogni cliente ha diverse sedi (ognuna con nome e indirizzo), e ogni consegna fa riferimento ad una specifica sede; ogni sede ha un numero, che la identifica univocamente fra quelle di uno stesso cliente.;
- per i prodotti sono di interesse (e debbono essere in qualche modo associabili alle rispettive bolle) i singoli esemplari, ognuno dei quali ha un numero di matricola (che identifica l’esemplare fra quelli del prodotto) e una data di produzione





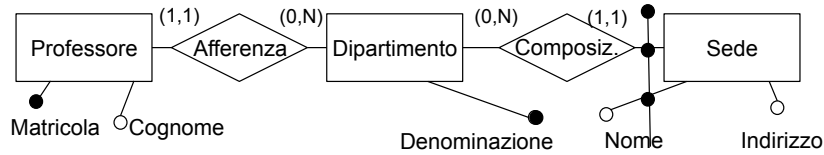
Possibili soluzioni

Tempo a disposizione: un'ora e quarantacinque minuti.

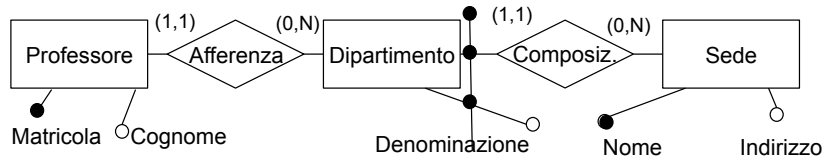
Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

**Domanda 1** (10%) Considerare i due schemi seguenti

a)



b)



Con riferimento a ciascuno di essi, rispondere alle seguenti domande, con un sì o un no negli spazi della tabella:

riordinare nei compiti B, C, D

	Schema (a)	Schema (b)
possono esistere due sedi con lo stesso nome per uno stesso dipartimento	NO	NO
possono esistere due dipartimenti per una sede?	SÌ	NO
possono esistere due sedi per un dipartimento?	NO	SÌ
possono esistere due sedi con lo stesso nome?	SÌ	NO
ad un professore è associato uno e un solo indirizzo?	NO	SÌ

**Domanda 2** (15%) Tradurre nel modello relazionale i due schemi ER mostrati nella domanda precedente, mostrando possibili istanze (che rendano evidenti le differenze fra le due realtà; bastano tre o quattro ennuple per ciascuna relazione). Indicare (ad esempio direttamente sulle tabelle) anche i vincoli di chiave e integrità referenziale.

a)  
 (scambiare con la successiva nei compiti B e D)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip )  
 - DIPARTIMENTO (Denominazione, ... )  
 - SEDE (DenominazioneDip, NomeSede, Indirizzo ... )

b)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip, SedeDip )  
 - DIPARTIMENTO (Denominazione, Sede ... )  
 - SEDE (NomeSede, Indirizzo ... )

**Domanda 3** (15%) Considerare il seguente schema relazionale (con vincoli di integrità referenziale fra l'attributo Prodotto di LINEEDORDINE e la chiave Matricola di PRODOTTI e fra l'attributo Ordine di LINEEDORDINE e la chiave Codice di ORDINI):

PRODOTTI			LINEEDORDINE			ORDINI		
Matricola	Nome	Prezzo	Prodotto	Ordine	Quantità	Codice	Data	Cliente
...	...	...	...	...	...	...	...	...

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità  $P = 10.000$  (PRODOTTI),  $O = 1.000$  (ORDINI) e  $L = 4.000$  (LINEEDORDINE), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT Matricola, Codice FROM Prodotti, Ordini	$P \times O$	$P \times O$	10.000.000	10.000.000
SELECT Prodotto, Ordine, Nome, Data FROM Prodotti JOIN LineeDOrdine ON Matricola=Prodotto JOIN Ordini ON Ordine = Codice	$L$	$L$	4.000	4.000
SELECT Prodotto, Nome, COUNT(*) FROM Prodotti JOIN LineeDOrdine ON Matricola=Prodotto GROUP BY Prodotto, Nome	$L/O$	$P$	4	10.000

Nota: Il valore  $L/O$  deriva dal numero minimo di prodotti che deve essere ordinato. Il valore deriva dal fatto che ogni ordine ha linee che riguardano prodotti diversi.

**Domanda 4** (25%) Considerare ancora lo schema mostrato nella domanda precedente.

1. Formulare in **SQL** l'interrogazione che trova le coppie di prodotti (è sufficiente mostrare gli identificatori) che compaiono insieme in almeno un ordine.

```
SELECT DISTINCT L1.Prodotto AS Prodotto1, L2.Prodotto AS Prodotto2
FROM LineeDOrdine L1, LineeDOrdine L2
WHERE L1.Ordine = L2.Ordine AND L1.Prodotto > L2.Prodotto
```

$$\pi_{P,P'}(\sigma_{P>P'}(\text{LINEEDORDINE} \bowtie_{O=O'} \rho_{X' \leftarrow X}(\text{LINEEDORDINE})))$$

2. Formulare in **algebra relazionale** l'interrogazione che trova i clienti che hanno ordinato almeno due prodotti (in uno stesso ordine o in ordini diversi). Mostrare, oltre ai clienti, le coppie di prodotti

Utile una vista

```
CREATE VIEW V AS SELECT *
FROM LineeDOrdine JOIN Ordini ON Ordine = Codice
```

```
SELECT V1.Cliente, V1.Prodotto, V2.Prodotto
FROM V AS V1 JOIN V AS V2 ON V1.Cliente = V2.Cliente
WHERE V1.Prodotto > V2.Prodotto
```

$$V := \text{LINEEDORDINE} \bowtie_{O=C} \text{ORDINI}$$

$$\pi_{P,P'}(\sigma_{P>P'}(V \bowtie_{C=C'} \rho_{X' \leftarrow X} V))$$

3. Formulare in **SQL** l'interrogazione che trova i prodotti (mostrando tutte le proprietà) che sono stati ordinati da un solo cliente (in uno o più ordini).

```
SELECT Prodotti.*
FROM Prodotti JOIN V V1 ON Matricola = Prodotto
WHERE NOT EXISTS (SELECT *
FROM V V2
WHERE V1.Cliente <> V2.Cliente AND V1.Prodotto = V2.Prodotto)
```

La domanda continua a pagina seguente.

Ripetiamo per comodità lo schema:

PRODOTTI			LINEEDORDINE			ORDINI		
Matricola	Nome	Prezzo	Prodotto	Ordine	Quantità	Codice	Data	Cliente
...	...	...	...	...	...	...	...	...

4. Formulare in SQL l'interrogazione che trova tutti gli ordini, indicando, per ciascuno, data, cliente, numero di prodotti diversi ordinati e importo totale (ottenuto attraverso la somma dei prodotti fra quantità e prezzo)

```
SELECT Codice, Data, Cliente, COUNT(*) AS NProdotti, SUM(Quantita*Prezzo)
FROM V JOIN Prodotti ON Prodotto = Matricola
GROUP BY Codice, Data, Cliente
```

**Domanda 5** (7.5%) Con riferimento ancora allo schema usato nelle domande precedenti, si vuole scrivere un metodo che stampi l'elenco dei prodotti, specificando per ciascuno l'elenco degli ordini in cui compare, con il cliente che lo ha ordinato e la relativa quantità. Si vogliono stampare anche i prodotti per i quali non c'è nessun ordine. Il frammento di codice che (una volta completato con l'interrogazione SQL passata come argomento alla `executeQuery` nella seconda riga) produce la stampa è il seguente.

```
Statement query = con.createStatement();
ResultSet result = query.executeQuery("                ");
// La riga precedente va completata con l'interrogazione SQL
int prodottoPrec = -1;
while (result.next()){
    int prodotto = result.getInt("prodotto");
    if (!(prodotto==prodottoPrec)) {
        System.out.println("Prodotto: " + prodotto);
        prodottoPrec = prodotto ;
    }
    int ordine = result.getInt("ordine");
    int quantita = result.getInt("quantita");
    String cliente = result.getString("cliente");
    System.out.println(" Ordine: " + ordine + " Cliente: " + cliente + ". Quantita:" +
        quantita);
}
```

Scrivere l'interrogazione SQL da inserire come argomento della `executeQuery` nella seconda riga.

```
SELECT Matricola AS Prodotto, Ordine, Cliente, Quantita
FROM Prodotti LEFT JOIN (LineeDOrdine JOIN Ordini ON Ordine = Codice)
    ON Matricola = Prodotto
ORDER BY Matricola, Ordine
```

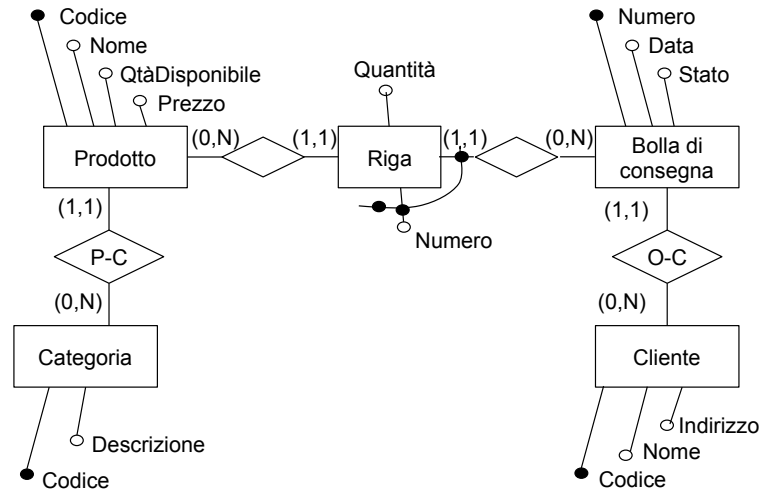
**Domanda 6** (7.5%) Definire (con una opportuna notazione) su una relazione

CONTICORRENTI(Matricola,Saldo,Interessi)

un vincolo che imponga che il valore di **Interessi** è positivo solo se quello di **Saldo** è almeno pari a 5.000.

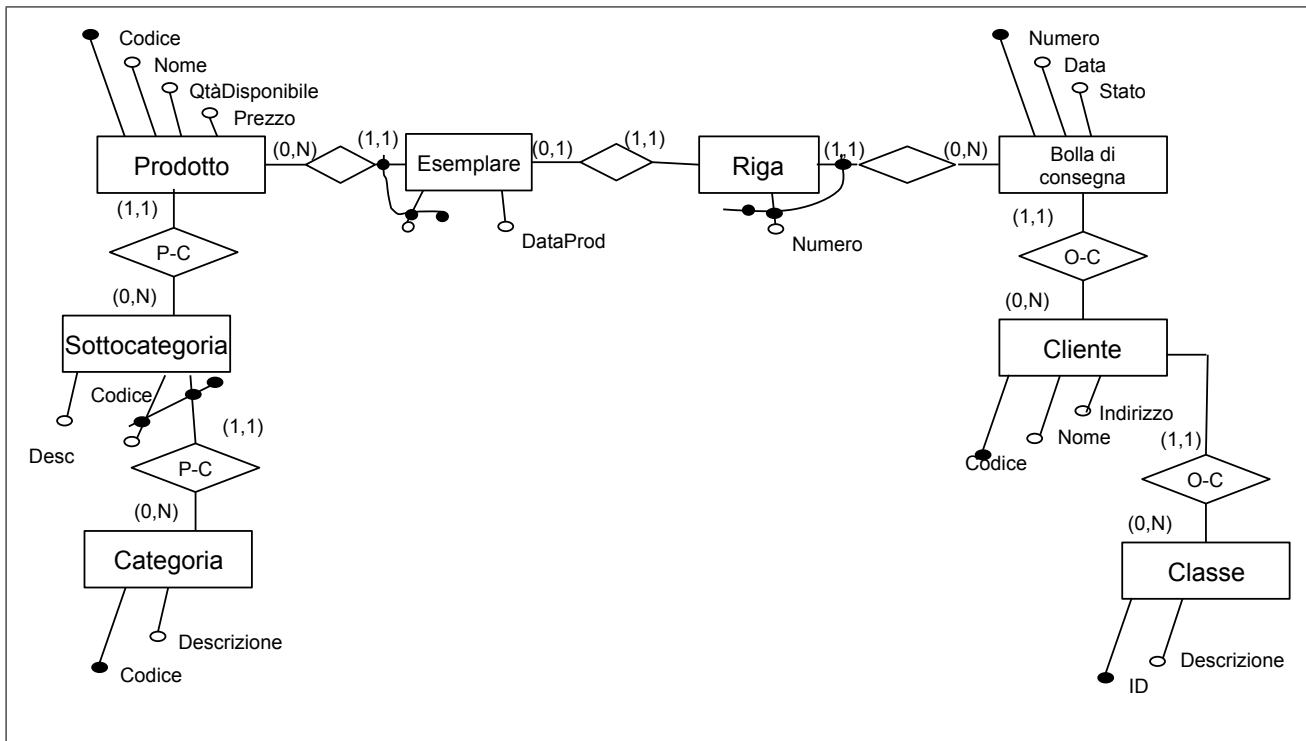
```
CHECK (NOT (Interessi > 0) OR Saldo > 5.000)
```

**Domanda 7 (20%)** Considerare il seguente schema concettuale, simile a quello discusso come soluzione del progetto proposto nel corso (la differenza principale è che, invece che di ordini, si tratta di bolle di consegna, i documenti che descrivono la fase di consegna dei prodotti):



Modificare lo schema sulla base delle specifiche seguenti

- i clienti sono raggruppati in classi (al momento, “privati,” “business” e “top business,” ma potrebbero cambiare nel tempo); ogni classe ha un identificatore e una descrizione;
- le categorie sono articolate in sottocategorie (anche loro con codice e descrizione) e ogni prodotto appartiene ad una sottocategoria; le sottocategorie hanno codici che sono univoci solo nell’ambito di ciascuna categoria;
- per i prodotti sono di interesse (e debbono essere in qualche modo associabili alle rispettive bolle) i singoli esemplari, ognuno dei quali ha un numero di matricola (che identifica l’esemplare fra quelli del prodotto) e una data di produzione



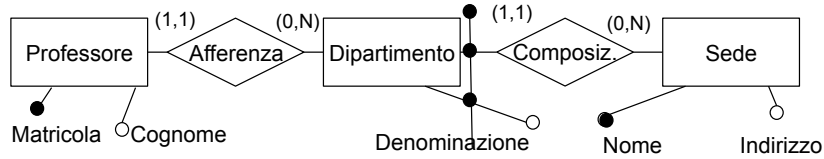
Possibili soluzioni

Tempo a disposizione: un'ora e quarantacinque minuti.

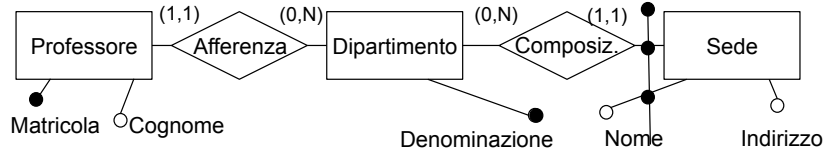
Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

Domanda 1 (10%) Considerare i due schemi seguenti

a)



b)



Con riferimento a ciascuno di essi, rispondere alle seguenti domande, con un sì o un no negli spazi della tabella:

riordinare nei compiti B, C, D

	Schema (a)	Schema (b)
possono esistere due sedi per un dipartimento?	NO	NO
possono esistere due sedi con lo stesso nome?	SÌ	NO
possono esistere due sedi con lo stesso nome per uno stesso dipartimento	NO	SÌ
possono esistere due dipartimenti per una sede?	SÌ	NO
ad un professore è associato uno e un solo indirizzo?	NO	SÌ

Domanda 2 (15%) Tradurre nel modello relazionale i due schemi ER mostrati nella domanda precedente, mostrando possibili istanze (che rendano evidenti le differenze fra le due realtà; bastano tre o quattro ennuple per ciascuna relazione). Indicare (ad esempio direttamente sulle tabelle) anche i vincoli di chiave e integrità referenziale.

a)  
 (scambiare con la successiva nei compiti B e D)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip )  
 - DIPARTIMENTO (Denominazione, ... )  
 - SEDE (DenominazioneDip, NomeSede, Indirizzo ... )

b)  
 - PROFESSORE (Matr, Cognome, DenominazioneDip, SedeDip )  
 - DIPARTIMENTO (Denominazione, Sede ... )  
 - SEDE (NomeSede, Indirizzo ... )

**Domanda 3** (15%) Considerare il seguente schema relazionale (con vincoli di integrità referenziale fra l'attributo Prodotto di LINEEDORDINE e la chiave Codice di PRODOTTI e fra l'attributo Ordine di LINEEDORDINE e la chiave Numero di ORDINI):

PRODOTTI			LINEEDORDINE			ORDINI		
Codice	Nome	Prezzo	Prodotto	Ordine	Quantità	Numero	Data	Cliente
...	...	...	...	...	...	...	...	...

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità  $P = 30.000$  (PRODOTTI),  $O = 1.000$  (ORDINI) e  $L = 6.000$  (LINEEDORDINE), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT Codice, Numero FROM Prodotti, Ordini	$P \times O$	$P \times O$	30.000.000	30.000.000
SELECT Prodotto, Ordine, Nome, Data FROM Prodotti JOIN LineeDOrdine ON Codice=Prodotto JOIN Ordini ON Ordine = Numero	$L$	$L$	6.000	6.000
SELECT Prodotto, Nome, COUNT(*) FROM Prodotti JOIN LineeDOrdine ON Codice=Prodotto GROUP BY Prodotto, Nome	$L/O$	$P$	6	30.000

Nota: Il valore  $L/O$  deriva dal numero minimo di prodotti che deve essere ordinato. Il valore deriva dal fatto che ogni ordine ha linee che riguardano prodotti diversi.

**Domanda 4** (25%) Considerare ancora lo schema mostrato nella domanda precedente.

- Formulare in **algebra relazionale** l'interrogazione che trova le coppie di prodotti (è sufficiente mostrare gli identificatori) che compaiono insieme in almeno un ordine.

```
SELECT DISTINCT L1.Prodotto AS Prodotto1, L2.Prodotto AS Prodotto2
FROM LineeDOrdine L1, LineeDOrdine L2
WHERE L1.Ordine = L2.Ordine AND L1.Prodotto > L2.Prodotto
```

$$\pi_{P,P'}(\sigma_{P>P'}(\text{LINEEDORDINE} \bowtie_{O=O'} \rho_{X' \leftarrow X}(\text{LINEEDORDINE})))$$

- Formulare in **SQL** l'interrogazione che trova i clienti che hanno ordinato almeno due prodotti (in uno stesso ordine o in ordini diversi). Mostrare, oltre ai clienti, le coppie di prodotti

Utile una vista

```
CREATE VIEW V AS SELECT *
FROM LineeDOrdine JOIN Ordini ON Ordine = Codice
```

```
SELECT V1.Cliente, V1.Prodotto, V2.Prodotto
FROM V AS V1 JOIN V AS V2 ON V1.Cliente = V2.Cliente
WHERE V1.Prodotto > V2.Prodotto
```

$$V := \text{LINEEDORDINE} \bowtie_{O=C} \text{ORDINI}$$

$$\pi_{P,P'}(\sigma_{P>P'}(V \bowtie_{C=C'} \rho_{X' \leftarrow X} V))$$

- Formulare in **SQL** l'interrogazione che trova i prodotti (mostrando tutte le proprietà) che sono stati ordinati da un solo cliente (in uno o più ordini).

```
SELECT Prodotti.*
FROM Prodotti JOIN V V1 ON Matricola = Prodotto
WHERE NOT EXISTS (SELECT *
FROM V V2
WHERE V1.Cliente <> V2.Cliente AND V1.Prodotto = V2.Prodotto)
```

La domanda continua a pagina seguente.

Ripetiamo per comodità lo schema:

PRODOTTI			LINEEORDINE			ORDINI		
Codice	Nome	Prezzo	Prodotto	Ordine	Quantità	Numero	Data	Cliente
...	...	...	...	...	...	...	...	...

4. Formulare in SQL l'interrogazione che trova tutti gli ordini, indicando, per ciascuno, data, cliente, numero di prodotti diversi ordinati e importo totale (ottenuto attraverso la somma dei prodotti fra quantità e prezzo)

```
SELECT Codice, Data, Cliente, COUNT(*) AS NProdotti, SUM(Quantita*Prezzo)
FROM V JOIN Prodotti ON Prodotto = Matricola
GROUP BY Codice, Data, Cliente
```

**Domanda 5** (7.5%) Con riferimento ancora allo schema usato nelle domande precedenti, si vuole scrivere un metodo che stampi l'elenco dei prodotti, specificando per ciascuno l'elenco degli ordini in cui compare, con il cliente che lo ha ordinato e la relativa quantità. Si vogliono stampare anche i prodotti per i quali non c'è nessun ordine. Il frammento di codice che (una volta completato con l'interrogazione SQL passata come argomento alla `executeQuery` nella seconda riga) produce la stampa è il seguente.

```
Statement query = con.createStatement();
ResultSet result = query.executeQuery("                ");
// La riga precedente va completata con l'interrogazione SQL
int prodottoPrec = -1;
while (result.next()){
    int prodotto = result.getInt("prodotto");
    if (!(prodotto==prodottoPrec)) {
        System.out.println("Prodotto: " + prodotto);
        prodottoPrec = prodotto ;
    }
    int ordine = result.getInt("ordine");
    int quantita = result.getInt("quantita");
    String cliente = result.getString("cliente");
    System.out.println(" Ordine: " + ordine + " Cliente: " + cliente + ". Quantita:" +
        quantita);
}
```

Scrivere l'interrogazione SQL da inserire come argomento della `executeQuery` nella seconda riga.

```
SELECT Matricola AS Prodotto, Ordine, Cliente, Quantita
FROM Prodotti LEFT JOIN (LineeDOrdine JOIN Ordini ON Ordine = Codice)
    ON Matricola = Prodotto
ORDER BY Matricola, Ordine
```

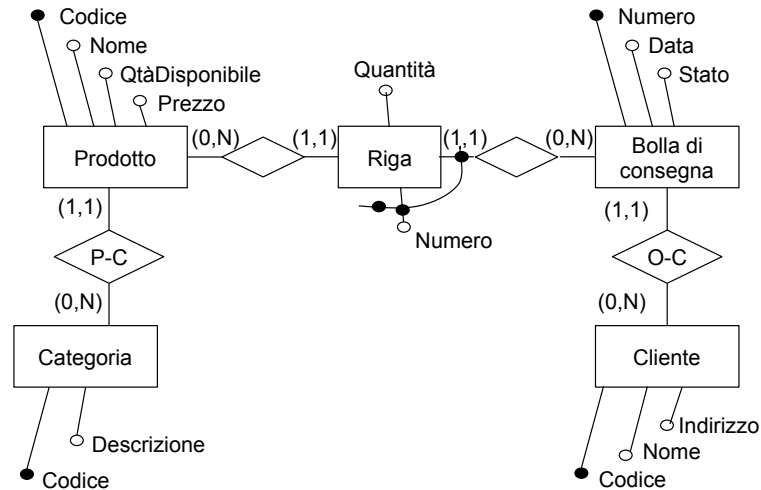
**Domanda 6** (7.5%) Definire (con una opportuna notazione) su una relazione

SOCI(Matricola,Punti,PuntiPremio)

un vincolo che imponga che il valore di `PuntiPremio` è positivo solo se quello di `Punti` è almeno pari a 1.000.

```
CHECK (NOT (PuntiPremio > 0) OR Punti > 1.000)
```

**Domanda 7 (20%)** Considerare il seguente schema concettuale, simile a quello discusso come soluzione del progetto proposto nel corso (la differenza principale è che, invece che di ordini, si tratta di bolle di consegna, i documenti che descrivono la fase di consegna dei prodotti):



Modificare lo schema sulla base delle specifiche seguenti

- i clienti sono raggruppati in classi (al momento, “privati,” “business” e “top business,” ma potrebbero cambiare nel tempo); ogni classe ha un identificatore e una descrizione;
- ogni cliente ha diverse sedi (ognuna con nome e indirizzo), e ogni consegna fa riferimento ad una specifica sede; ogni sede ha un numero, che la identifica univocamente fra quelle di uno stesso cliente.;
- per i prodotti sono di interesse (e debbono essere in qualche modo associabili alle rispettive bolle) i singoli esemplari, ognuno dei quali ha un numero di matricola (che identifica l’esemplare fra quelli del prodotto) e una data di produzione

